



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/783,842

02/20/2004

Samuel W. Bent

MS1-1955US

7498

22801

7590

07/01/2008

LEE & HAYES PLLC

421 W RIVERSIDE AVENUE SUITE 500

SPOKANE, WA 99201

EXAMINER

YIGDALL, MICHAEL J

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

07/01/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/783,842	<b>Applicant(s)</b> BENT ET AL.	
	<b>Examiner</b> Michael J. Yigdall	<b>Art Unit</b> 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 14 May 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1,4-11,13,14,17-26,28,32-41 and 43-47 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,4-11,13,14,17-26,28,32-41 and 43-47 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)            | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>05/14/2008</u> .  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on May 14, 2008 has been entered. Claims 1, 4-11, 13, 14, 17-26, 28, 32-41 and 43-47 are pending.

### ***Response to Arguments***

2. Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that in Bent, a "format object" merely "dictates how the data is represented," while as recited in claim 1, a "transform definition" changes "how the data item value is displayed in [a] ... user interface" (remarks, page 26, paragraph [0025]).

However, the examiner respectfully notes that the format objects of Bent do, in fact, change how data item values are displayed in a user interface. Bent describes that the format objects control "how data is presented" (column 16, line 31). The format objects provide properties, methods and events that allow the end user to change how the data is "presented on a display screen" (column 15, lines 27-34). In other words, the format objects change how the data is displayed in a user interface. Bent clearly describes "data conversion and formatting to control user interface displays in the context of data binding" (abstract).

Applicant refers to a "data style definition" that defines "a visual representation of the data item on a user interface" (remarks, page 26, paragraph [0026]) and a "content presenter"

configured to apply a “data style definition to an instantiation of a display element on the user interface” to display data items “according to the defined visual representation by combining the data from the data items and the information from the data style definition” (paragraph [0027]) as elements of claim 1 that are not found in Bent.

However, the examiner respectfully notes that the last Office action addressed such elements in dependent claims 12 and 16, for example. The format objects of Bent provide formatting options that define the visual representation of the data on the user interface (see, for example, column 16, lines 3-24). In other words, the formatting options represent a data style definition. Furthermore, Bent describes that the binding system is configured to apply the format objects (and thus the data style definition) to elements of the user interface (see, for example, column 15, lines 35-49). The data from the data store is combined with the formatting options from the data style definition and displayed on the user interface (see, for example, column 15, lines 59-67). Thus, Bent teaches a content presenter such as recited in the claim.

In response to Applicant’s arguments in support of claims 17 and 32 referring to the “content presenter” element (remarks, page 30, paragraph [0038] and page 33, paragraph [0047]), the examiner respectfully notes that claims 17 and 32, as amended, do not recite a content presenter.

3. Applicant’s other arguments have been considered but are moot in view of the new ground(s) of rejection set forth below.

### ***Claim Rejections - 35 USC § 112***

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

Art Unit: 2192

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 13 and 32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicant regards as the invention.

With respect to claim 13 (currently amended), the claim recites “a tree assembler module comprising a style lookup module and the data bind module.” However, there is insufficient antecedent basis for “the data bind module” in claims 1 and 13. The examiner’s interpretation is that claim 13 was intended to introduce “a data bind module.”

With respect to claim 32 (currently amended), the claim recites “using a tree assembler module comprising a style lookup module and the data module.” However, there is insufficient antecedent basis for “the data bind module” in the claim. The examiner’s interpretation is that the claim was intended to introduce “a data bind module.”

### ***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1, 4-11, 13, 14, 17-26, 28, 32-41 and 43-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,463,442 to Bent et al. (art of record, “Bent”) in view of U.S. Pub. No. 2003/0025728 to Ebbo et al. (now made of record, “Ebbo”).

With respect to claim 1 (currently amended), Bent teaches a computer readable storage medium having a data programming model executable by a processor embodied thereon, the data programming model (see, for example, FIG. 1 and the abstract), comprising:

a data item having a plurality of data item properties, each of these data item properties having an associated value (see, for example, data 250 in FIG. 3, and column 9, lines 8-14, which shows a data source or data item with properties and associated values);

a user interface element having an element property with a value that can be defined by an association to a respective data item property (see, for example, display areas 210 and 220 in FIG. 3, and column 6, lines 32-36, which shows user interface elements that are data consumers, and see, for example, column 8, lines 44-49, which further shows the data consumer with a property that defines an association to the data source or data item property);

a data style definition configured to define a visual representation of the data item on a user interface, wherein the associated values of the data item properties influence the appearance of a plurality of features of the user interface element, wherein the data items are maintained independently from the data style definition (see, for example, column 16, lines 3-24, which shows data formatting options or a data style definition that defines a visual representation of the data item on the user interface, and column 15, lines 59-67, which shows that the data item is independent of the data style definition).

Bent further teaches arranging data style definitions in a tree as a collection of data style definitions (see, for example, column 19, lines 10-22), but does not explicitly disclose that the data item is represented as a subtree of the user interface element and the data item properties are represented as part of the subtree.

However, in an analogous art, Ebbo teaches binding data items to user interface elements and representing the user interface elements as a hierarchy of control objects (see, for example, paragraph [0047]). The hierarchy comprises at least one subtree of control objects (see, for example, paragraph [0046]).

One of ordinary skill in the art could, with predictable results, represent the data item of Bent as a subtree of the user interface element and the data item properties as part of the subtree. Therefore, as Ebbo suggests, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Bent such that the data item is represented as a subtree of the user interface element and the data item properties are represented as part of the subtree.

Bent in view of Ebbo further teaches:

a binding definition configured to associate the element property of the user interface element with the data item property (see, for example, column 7, lines 46-57, which shows a binding definition that associates the user interface element with the data source or data item);

a transform definition developed as a logic component of an application program, the transform definition configured to generate a transformed value of the data item property for association with an element property of the user interface element (see, for example, column 15, lines 27-34, which shows a format object that provides a transform definition to generate a

Art Unit: 2192

formatted or transformed value for association with the user interface element, and column 16, lines 25-33, which further shows a transform definition developed as a component of the application program), wherein the transform definition is interjected in a data path between the data item and the user interface, wherein the transform definition changes how the data item value is displayed in the user interface (see, for example, column 16, lines 25-33, which shows that the transform definition is interjected into the data transfer path and changes how the data item is presented in the user interface element), wherein presentation of the data item is dependent upon the transform definition, wherein the presentation characteristics are not integrated with the data item and are integrated with the data style definition (see, for example, column 15, lines 59-67, which shows that the presentation of the data item depends on the transform definition and that the presentation characteristics of the format object are not integrated with the data item, and column 16, lines 3-24, which shows that the presentation characteristics are integrated with the data style definition), and wherein the transformed value of the data item property being generated from the associated value of the data item property such that the associated value of the data item property is maintained unchanged in a data item database (see, for example, data store 141 in FIG. 3, and column 15, lines 59-67, which shows that the associated value is maintained unchanged in a data store or database); and

a content presenter configured to apply the data style definition to an instantiation of a display element on the user interface to display one or more of the data items according to the defined visual representation by combining the data from the data items and the information from the data style definition (see, for example, column 15, lines 35-49, which shows a content presenter configured to apply the data formatting options or data style definition to the user



interface element to display the data item, and column 15, lines 59-67, which shows that data from the data store is combined with the formatting options from the data style definition).

With respect to claim 4 (previously presented), the rejection of claim 1 is incorporated. Bent does not explicitly disclose one-time binding wherein a user interface property is initialized from a data item property and wherein the user interface property does not update when changes are made to the data item property after the initialization.

Nonetheless, Bent teaches a “dataMemberChanged” method that notifies a data consumer when changes are made to a data item after initialization, and discloses that it is necessary to call a “getDataMember” method to actually update the data consumer with the changes (see, for example, column 10, lines 56-63). Thus, one of ordinary skill in the art could, with predictable results, implement the teachings of Bent such that the “getDataMember” method is not called after initialization, in which case the data consumer would not update when changes are made to the data item. Furthermore, Bent teaches a “removeDataSourceListener” method that removes a data consumer from the notification list, such that the data consumer would not know when changes are made to a data item (see, for example, column 10, lines 47-55). The method thus “facilitates” one-time binding.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the teachings of Bent such that the binding definition facilitates one-time binding wherein a user interface property is initialized from a data item property and wherein the user interface property does not update when changes are made to the data item property after the initialization.

With respect to claim 5 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches a transform definition configured to generate a transformed value of the data item property for compatible association with the element property of the user interface element (see, for example, column 15, lines 27-34, which shows a format or transform definition to generate a formatted or transformed value for association with the user interface element).

With respect to claim 6 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches an update logic component configured to receive a data item update that corresponds to a change of the associated value of the data item property, and further configured to initiate that the binding definition update the element property of the user interface element with the associated value change of the data item property (see, for example, column 12, lines 2-5, which shows an update component that receives a change in the value of the data source or data item and updates the user interface element).

With respect to claim 7 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches an update logic component configured to receive a user interface element update that corresponds to a change of the value of the element property of the user interface element, and further configured to update the associated data item property with the value change of the element property of the user interface element (see, for example, column 11, lines 56-61, which shows an update component that receives a change in the value of the user interface element and updates the data source or data item).

With respect to claim 8 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches:

a data context property configured to define the data item as the data source of the user interface element (see, for example, column 12, lines 11-16, which shows a collection or context property that defines the data source of the user interface element); and

an additional binding definition configured to associate an element property of an additional user interface element with an additional data item property of the data item, the additional user interface element having a dependent association to the user interface element, and the additional binding definition further configured to default to the data context property to define the data item as the data source of the additional user interface element (see, for example, column 20, lines 46-55, which shows an additional binding definition for an additional user interface element in the form of a compound user interface element with such a dependent association, and column 20, lines 62-67, which further shows that the collection or context property sets the default data source for the additional user interface element).

With respect to claim 9 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches:

a data context property configured to define the data item as the data source of the user interface element (see, for example, column 12, lines 11-16, which shows a collection or context property that defines the data source of the user interface element);

an additional binding definition configured to associate an element property of an additional user interface element with an additional data item property of the data item, the additional user interface element having a dependent association to the user interface element,

and the additional binding definition further configured to default to the data context property to define the data item as the data source of the additional user interface element (see, for example, column 20, lines 46-55, which shows an additional binding definition for an additional user interface element in the form of a compound user interface element with such a dependent association, and column 20, lines 62-67, which further shows that the collection or context property sets the default data source for the additional user interface element); and

wherein a change of the value of the element property of the user interface element initiates a change of a value of the element property of the additional user interface element according to the default data context property (see, for example, column 20, lines 56-59, which shows that the additional user interface element reflects such changes).

With respect to claim 10 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches:

a collection of data items (see, for example, column 8, lines 32-39, which shows a collection of data sources or data items); and

a representation of the data items each configured for display in a user interface display element that is associated with a referenced data item in the representation of the data items (see, for example, column 20, lines 32-40, which shows a representation of the data sources or data items displayed in associated user interface elements).

With respect to claim 11 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches:

a collection of data items (see, for example, column 8, lines 32-39, which shows a collection of data sources or data items); and

a first representation of the data items in the collection and at least a second different representation of the data items in the collection, the first representation and the second different representation each being configured to reference the data items in the collection (see, for example, column 21, lines 16-29, which shows different representations of the data sources or data items).

With respect to claim 13 (currently amended), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches that the data style definition further comprises an internal representation of the user interface element, wherein the user interface element represents the combination of the value of the data item property and information from the data style definition, the data style definition describing how to display the data item property (see, for example, column 16, lines 3-24, which shows that the data style definition describes how to display the data item property and comprises formatting options that represent an internal representation of user interface element, and see, for example, column 15, lines 27-34, which shows that user interface element represents the combination of the data from the data store and the formatting options from the data style definition);

a template, wherein the template is a general form of the visual representation of the data item, wherein the template requires that part of the visual representation be obtained as a data item property value (see, for example, column 20, lines 7-27, which shows an example formatting option in the general form “\$#, ##0.00” that represents such a template); and

a tree assembler module comprising a style lookup module and the data bind module, wherein the data bind module locates any style elements of the data item and binds a property of a user interface element to a data item property (see, for example, binding agent 310 in FIG. 3, and column 7, lines 18-20, which shows binding a property of a user interface element to a data item property, and see, for example, column 15, lines 35-49, which further shows locating any style elements).

Note that while Bent does not explicitly disclose that the style lookup module and data bind module form a “tree assembler” module, Ebbo further teaches that because the user interface elements are represented as a hierarchy of control objects, the user interface elements are rendered with a tree traversal sequence (see, for example, paragraph [0038]), and therefore such a module would have been obvious to those of ordinary skill in the art at the time the invention was made.

With respect to claim 14 (original), the rejection of claim 1 is incorporated, and Bent in view of Ebbo further teaches a data style definition configured to define a visual representation of data items as a data tree (see, for example, Ebbo, paragraphs [0046] and [0047], and see the rejection of claim 1 above).

With respect to claims 17-24 (currently amended), 25 (original), 26 and 28 (currently amended), the claims are directed to a computing system comprising elements and features that were previously addressed with respect to claims 1, 4-11, 13 and 14 (see the rejection of claims 1, 4-11, 13 and 14 above).

With respect to claims 32 (currently amended), 33-41 (original) and 43-46 (currently amended), the claims are directed to a method comprising steps and features that were previously addressed with respect to claims 1, 4-11, 13 and 14 (see the rejection of claims 1, 4-11, 13 and 14 above).

With respect to claim 47 (currently amended), the rejection of claim 32 is incorporated, and Bent in view of Ebbo further teaches declaring an instance of a data class which corresponds to a type of data as a resource, and wherein defining the binding association comprises referring to the data class in a declaration of the binding association (see, for example, column 7, lines 46-57, which shows declaring a data source object or an instance of a data class and referring to the data source object to define the binding association).

### ***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is 571-272-3707. The examiner can normally be reached on Monday to Friday from 8:00 AM to 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael J. Yigdall  
Examiner  
Art Unit 2192

/Michael J. Yigdall/  
Examiner, Art Unit 2192